REMARKS

In the Office Action mailed June 2, 2009 and the Advisory Action of October 23, 2009, the Office noted that claims 22-41 and 43-46 were pending and rejected claims 22-41 and 43-46. Claim 32 has been amended, no claim has been canceled, and, thus, in view of the foregoing claims 22-41 and 43-46 remain pending for reconsideration which is requested. No new matter has been added. The Office's rejections and objections are traversed below.

DRAWINGS

The Applicant acknowledges that the drawings submitted on March 16, 2009, were not entered.

OBJECTION TO THE SPECIFICATION

The disclosure stands objected to for informalities. In particular, the Office asserts that the Specification introduces new matter.

While the Applicant amended the Specification in the amendment of October 2, 2009, he now re-introduces the use of the word "map" in place of "card." The Applicants submit that no new matter is believed to have been added for the following reasons:

In the instant application the use of the word "card" resulted from an obvious error of translation.

The French word *"carte"* used in the international application (see the paragraph beginning at page 3, line 6 of

PCT/FR03/03877) has several meanings in English among which:

- a "map" (such as a "road map" for *"carte routière"*) or a "chart" (such as a "nautical chart" for *"carte nautique"*);
- a "card" (such as a "business card" for *"carte de visite"* or a "smart card" for *"carte à puce"*).

¶ 0018 of the printed publication version of the Specification wouldn't mean anything if read as "…provision is made to establish, in a secure environment, a **card** with the memory context of the authentic software application…". What is established in step 1) is obviously a "**map**" of the memory context (as defined in ¶ 0017) with the addresses in the memory of the symbols, of the instructions, of the data… when the authentic software is actually executed in a secure environment. A "card" cannot be established with a memory context, whereas a "map" can.

To make this clear find attached French / English dictionaries illustrating both meanings of the French / English words *"carte"* / "map" in the field of computer sciences.

In Harrap's (appendix 1), it is said that "memory map" means *"programme (de la) mémoire"* in French. It is not a very good translation (we would like better *"carte (de la) mémoire"*), but it shows that a "memory map" is not a "memory card" since it has to do with a program and not with the support of this program.

- In Routledge (appendix 2), it is said that a *"carte de mémoire"* can be either a "memory card" (we do not agree

with that) or a "memory map", but that a *"carte à mémoire"* is a "smart card". *"Carte de mémoire"* and *"carte à mémoire"* clearly have two different meanings in French.

Thus, the amended Specification and through it the claims find support in the parent Application and do not introduce new matter.

Further, there is no card in the general definition of the invention (¶¶ 0007-0010 of printed publication version of the Specification, as well as claims 1 and 34). There is only a memory context which implies a map of this context for one of ordinary skill in the art in the field of the claims.

There is only a smart card in the particular embodiment of figure 2, claim 35. Claim 35 clearly shows that this card contains the instructions of the certificate established by the processing means of claims 34 on the basis of the map (wrongly called "card") disclosed in ¶ 0018.

One of ordinary skill in the art cannot be confused about that. It should be noticed that the memory context is not only known by ¶ 0017 of the printed publication version of the Specification; it is a notion generally well known by computer scientists, for instance in the field of context switching where saving the memory map of the context of the running program and loading the memory map of the context of the next program to be run are essential steps.

Withdrawal of the objections is respectfully requested.

MEANS PLUS FUNCTION

The Applicant acknowledges that the Office considers claims 34-36, 39, 41, 42, and 44 to satisfy the requirements of 35 U.S.C. § 112, sixth paragraph.

REJECTIONS under 35 U.S.C. § 112

Claims 30, 34-42 and 44 stand rejected under 35 U.S.C. § 112, second paragraph as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. In particular the Office asserts that the claims contain antecedent basis issues and that the means plus function of some of the claims are not supported by the Specification. The Applicant has amended the claims to remove any antecedent basis issues.

The Office asserts that there is no structural support for "means for moving said executable certificate to the host terminal," as in claim 34.

However, ¶ 0034 of the printed publication version of the Specification states

> With reference to FIG. 1, the software application to be verified 1 is loaded into a host terminal (not illustrated). For example, the host terminal belongs to the group formed by ***data processing apparatuses, digital television decoders, equipment for visualising multimedia contents, micro-computers, smart cards, personal organisers, game consoles, mobile telephones or the like***. [Emphasis added]

Moreover, such a means can also be electromagnetic waves (digital television decoders, mobile telephones), DVDs

(equipment for visualizing multimedia contents, micro-computers) or any other convenient means.

The Office asserts that there is no structural support for "comparison means for positively comparing the result obtained through the execution of the control instructions with the result expected from an authentic application," as in claim 34.

However, ¶ 0011 of the printed publication version of the Specification states

> Understood here by the term "positive comparison" is the fact that **any action, operation or modification on the data used by the software application to be verified or any action, operation or modification on the running of the execution of the software application** to be verified produces a behaviour of the software application to be verified identical to that which is expected by the running of the execution of the authentic application. [Emphasis added]

Further ¶ 0036 of the printed publication version of the Specification states

> These protected data 2 contain 7 an executable certificate 4 including a series of non-protected control instructions, **which are executed 5 by the application to be verified 1. In practice, the control instructions for the executable certificate 4 are coded in the language of the processor of the host terminal, still called machine language.** As a variant, the instructions for the executable certificate 4 may also be coded in the language of a virtual machine, emulating the behaviour of a processor. [Emphasis added]

It is respectfully submitted that one skilled in the art would understand the program when executed on the host terminal would perform the comparison (i.e. a well know machine level instruction). Thus, the comparison means has structural

support in the machine level language performing the comparison.

The Office asserts that there is no structural support for "means which are capable, in the event of a positive comparison, of continuing with the execution of the software application to be verified," as in claim 34.

However, as argued above, the program executing the control instructions and performing comparison to determine further execute on is found in the machine language executing the program.

The Office asserts that there is no structural support for "a means for communicating with the secure circuit," as in claim 35.

It is respectfully submitted that the card reader is such a means.

The Office asserts that there is no structural support for "means which is capable of validating or invalidating the authenticity of the software application," as in claim 36.

However, it is respectfully submitted that such authentication is support by the program as argued with reference to ¶¶ 0011 and 0036 as argued above.

The Office asserts that there is no structural support for "means which are capable of inserting the executable certificate into a first stream of data," as in claim 44.

However, ¶ 0039 states

The **executable certificate** 4 may also be inserted in

> the **stream of data which the application 1 is capable of processing.** The insertion of executable certificates in a stream of data may correspond to the case where it is necessary to authenticate a protected multimedia stream processing application, accessible to the user on condition that the latter has fulfilled obligations such as defined by the seller of the contents. T**he source of the multimedia stream may be a transmission point of a broadcasting network**, the permanent memory of the host terminal, or even a memory unit extractable from the host terminal.

Thus, the executable certificate may be inserted by the transmission point of the broadcasting network. It is respectfully submitted that there is structure for the means which are capable of inserting the executable certificate into a first stream of data.

Withdrawal of the rejections is respectfully requested.

REJECTIONS under 35 U.S.C. § 102

Claims 22-23, 25-41 and 43-46 stand rejected under 35 U.S.C. § 102(e) as being anticipated by McCarroll, U.S. Patent Publication No. 2003/0196102. The Applicant respectfully disagrees and traverses the rejection with an argument.

On page 2 of the Advisory Action, the Office states "[i]n response to the applicant's argument that the signature disclosed by McCarroll is not a certificate according to the invention since it cannot be executed, the Office respectfully points out that the portion of the software code for the game that is digitally signed is being read on the executable certificate and not the accompanying signature file containing a

digest that corresponds to an unmodified portion of software code for the game that is digitally signed."

However, as best understood the Office asserts that McCarroll is equivalent to the executable certificate of the claims is the portion of the code that is signed and not the resulting signature. Therefore, it would appear that the Office acknowledges that the signature is not executable.

Thus, the portion of the code that is signed is not determined using the memory context of the authentic software application during the course of execution. This portion of the code, which can be found in the Specification at ¶ 0026 is determined by the programmer when drafting the code, obviously not during its execution, using the specifications, his personal knowledge and whatever else he may need, but certainly not the memory context.

Further, "memory context" has no meaning for a program which in not running, *a fortiori* for a program which is not completed. This is made clear in ¶ 0017 which mentions as elements of the memory context the addresses in the memory of a number of elements which are in the memory only during the execution of the program.

Further, claim 22 recites "using the memory context of…for determining at least…" which implies an actual action, whereas, in McCarroll the portion of the code is merely selected.

In the Advisory Action, the Office further states

> In response to the applicant's argument that McCarroll performs a mathematical comparison and not a positive comparison as defined in the specification, the examiner respectfully points out that a positive comparison is defined as "any action, operation or modification on the data used by the software application to be verified or any action, operation or modification on the running of the execution of the software application to be verified produces a behavior of the software application to be verified identical to that which is expected by the running of the execution of the authentic application." See applicant's specification page 2 lines 18-23. The examiner now points out that hashing a portion of software code into a first message digest is any action, operation or modification, the first message digest is a behavior of the portion of software code, and a second message digest from the signature file is that which is expected by running the execution of the authentic application. See McCarroll paragraph 30. Accordingly, the digest comparison performed by McCarroll fits within the definition of positive comparison defined in the applicant's specification.

The Applicant acknowledge that hashing a portion of software code is indeed an action. However, it t cannot be seriously maintained that "the first message digest is a behavior of the portion of software code."

A behavior is "the way in which something behaves" that is the way in which something "act, function or react" ; it is also "the manner of conducting oneself" (See Websters Dictionary).

A more technical definition of a software behavior in the field of computer science is given in the enclosed IEEE paper as "any changes, influences or any operations made to the other independent entities when the software works as an independent entity." *See appendix, Dynamic Trustiness Authentication Framework Based on Software's Behavior Integrity.*

In any case, a "behavior" is the way in which an entity interacts with itself or with its environment.

On the contrary, the first message digest is the result of hashing a portion of software code, it is another arrangement of the portion of code: it is an object, an entity. And, **an object cannot be a behavior.** An object has a behavior. An object and a behavior pertain to two different logical categories.

Further, what is expected by the running of the execution of the authentic application is also a behavior (see the construction of the sentence cited by the Office: *"…produces a behavior…identical to that which is expected…."* Therefore, the second message digest, also an object, cannot be either what is expected.

Therefore, the positive, behavioral, comparison as defined in the application is completely different from the structural comparison between the entities "first message digest" and "second message digest" of McCarroll.

The Office in the Advisory Action further states

> In response to the applicant's argument that McCarroll contains a tamperproof circuit for cryptographic operation whereas the certificate of the invention is executed on the host terminal itself, the examiner respectfully points out that the tamperproof circuit for cryptographic operations is contained in the host terminal. See McCarroll fig.1 ref. nos. 100 and 120. The examiner further points out that the claim language recites the transitional phrase "comprising," and therefore is inclusive or open-ended and does not exclude additional, unrecited elements or method steps.

However, it would appear that the interpretation of the

word "on" in the phrase "on the host terminal, executing the software application," is given undue weight

When interpreting a word in a text dealing with a specific technical field, one must use the technical meaning of the word. To execute a program on a terminal, means to have the CPU of the terminal executing the program. If a cell phone rests "on" a computer, nobody would say that the programs which are executed by the cell phone are executed "on" the computer. They are executed "on" the cell phone (computer science meaning) though "in" the cell phone (geographical meaning), but the computer is not concerned.

In McCarroll, the system 100 includes a CPU 110 and a tamperproof circuit 120. The tamperproof circuit 120 has its own processing (cryptography) unit 122. The cryptographic operations and the comparison between the first and second message digests are operated in the unit 122, otherwise the circuit 120 would not be tamperproof. Therefore, the so-called (by the Office) executable certificate is not executed on the host terminal (CPU 110), but on the tamperproof circuit 120.

It would appear that the Office considers that the system 100 is the host terminal of the claims, and not the CPU 110. The Applicant submits the term "terminal" obviously refers to the equivalent of the CPU of the system of McCarroll, not to the processing unit of a separate circuit geographically contained within the system but functionally distinct as the word "tamperproof" implies.

No matters the wording "comprising" in the claims. The actual means or steps which are claimed in the present application are not disclosed in McCarroll. There is no step in McCarroll of "using the memory context of the authentic software application during the course of execution" (see above), "on the host terminal, executing the software application to be verified" as it has just been showed, and of "positively comparing" (see above).

In the Advisory Action, the Office states

> In response to the applicant's argument that a signature is decrypted before the application is executed, the examiner respectfully disagrees with the applicant's position. The examiner respectfully points out that the execution of the software code for a game begins with the boot process and reading the software code from the game disc. The decryption of a signature occurs during this boot process and therefore the decryption of a signature is a step contained with the execution of the software code for game.

However, the definition of the beginning of the execution of the program given by the Office is purely personal and refers to no cited authority. Further, the Office's construction of the word "execution" runs counter McCarroll's teaching and even counter McCarroll's wording.

No where in McCarroll's is it stated that "execution" of the code begins with the boot process and the reading of the software code from the game disc. The execution of software begins when the first instruction of the software is executed by the CPU, not when the operating system of the host computer reads the program.

Further, McCarroll concerns a method to provide security. If the execution begins prior to the (non positive)

comparison of the message digests, there is no security since the execution happens outside of the tamperproof circuit 120. The boot process is therefore a matter of the operating system, not of the program.

Further, the terms of McCarroll's appear sufficiently clear. McCarroll ¶ 0029 mentions that the operation of the system 100 is prevented if the code is not valid. If the operation of the system is prevented, the program cannot be executed. There is no execution of the program before the comparison. In particular ¶ 0029 states "Thus, in the scenario where the code is not valid, the system 100 console would attempt to load the doctored disc 108, find that the signature is invalid, and refuse to boot the disc 108. This way, if somebody tries to modify the software on the disc 108, the software will not run on the system 100." (emphasis added).

Thus, in McCarroll, the sequence of operations is thus the following:

the code and the signature are received by the system;

the validity of the signature is determined by the cryptography unit;

if the signature is valid, the operation of the system continues: the disc is booted and the code is loaded and executed (run);

if the signature is invalid, the system is prevented from booting the disc and to load the code.

Thus, as best understood, the Office asserts that the code would be executed while it would not yet have been run. It is respectfully submitted that "dealing" with the disc is different that the execution of the code stored on the disc.

Claim 34 recites similar features as claim 22. Therefore, for at least the reasons argued above, claims 22 and 34 and the claims dependent therefrom are not anticipated by McCarroll.

Withdrawal of the rejections is respectfully requested

REJECTIONS under 35 U.S.C. § 103

Claim 24 stands rejected under 35 U.S.C. § 103(a) as being obvious over McCarroll in view of Yach, U.S. Patent Publication No. 2004/0025022. The Applicant respectfully disagrees and traverses the rejection with an argument.

Yach adds nothing to the deficiencies of McCarroll as applied against the independent claim. Therefore for at least the reasons discussed above, McCarroll and Yach, taken separately or in combination, fail to render obvious the features of claim 24.

SUMMARY

It is submitted that the claims satisfy the requirements of 35 U.S.C. §§ 112, 102 and 103. It is also submitted that claims 22-41 and 43-46 continue to be allowable. It is further submitted that the claims are not taught, disclosed or suggested by the prior art. The claims are therefore in a condition suitable for allowance. An early Notice of Allowance is requested.

The Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 25-0120 for any additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. § 1.17.

Respectfully submitted,

YOUNG & THOMPSON

/James J. Livingston/

James J. Livingston, Jr.
Reg.No. 55,394
209 Madison St, Suite 500
Alexandria, VA 22314
Telephone (703) 521-2297
Telefax (703) 685-0573
(703) 979-4709

JJL/jr/lad

APPENDIX:

The Appendix includes the following item(s):

☒ - a French/English dictionary

☒ - article, "*Dynamic Trustiness Authentication Framework Based on Software's Behavior Integrity.*"